TSX-Plus
Installation Guide

Fourth Edition
First Printing -- September, 1984

S&H Computer Systems, Inc.
1027 17th Avenue South
Nashville, Tennessee USA
37212-2299
(615)-327-3670

INTRODUCTION

The purpose of this guide is to help you install TSX-Plus on your system in the quickest and easiest way possible. This guide describes the contents of a typical distribution kit, necessary system resources, the circumstances that require a TSX-Plus system generation, the short-cut procedure for installing a functional version of TSX-Plus, and information about how to perform a full system generation to customize TSX-Plus to your needs.

The new TSX-Plus user should follow the short-cut path described in Chapter 2 to verify that all the components of the system are working together correctly and to obtain some experience with TSX-Plus before tailoring TSX-Plus to the peculiarities and needs of a particular site. The experienced TSX-Plus user may proceed directly to Chapter 3 for detailed instructions on performing the TSX-Plus system generation.

This guide is intended for the person installing TSX-Plus and assumes some familiarity with the commands and utilities of RT-11, including: handling of magnetic media; use of command files; the COPY and PRINT commands; use of one of the system editors (KED, EDIT, or TECO); assembling and linking a pre-written MACRO program; and the R[UN] command. It is not necessary to be able to write MACRO programs, but you should be familiar with the assembly and linking process. If you elect to perform a TSX-Plus system generation, it is also useful to be familiar with the usage of comments, labels and parameters in MACRO programs.

It is necessary to have access to configuration information about the various peripheral devices attached to your computer, especially to know what device handlers are needed and what vectors and addresses are used by these devices and by time-sharing terminals. This configuration information should have been written down and left with the system by the person who installed the hardware. If it is not available, ask the person who installed the hardware to enter the configuration information for your system in the form in Appendix B of this manual.

The TSX-Plus operating system is designed to provide the familiar RT-11 operating environment to multiple users. This goal mandates several extensions to the command language of RT-11, and makes a few features unavailable. The TSX-Plus Reference Manual describes all of the functional differences between RT-11 and TSX-Plus, including additional keyboard commands and other features, unsupported commands and other features, and other minor variations between the two operating systems.

The TSX-Plus manuals are intended to be used in conjunction with the RT-11 manuals; refer to the RT-11 manuals for descriptions of features which are not discussed in the TSX-Plus Reference Manual.

1. GETTING STARTED

1.1 Distribution Kit

The TSX-Plus distribution package you have received should contain the following items:

1.* TSX-Plus Reference Manual which describes the features of TSX-Plus.

2.* TSX-Plus System Manager's Guide which provides information needed by the system administrator.

3.* TSX-Plus installation guide.

* Note that only new orders include these three manuals. They are not automatically included with updates.

4. TSX-Plus release notes

5. A magnetic medium (reversible RX01 diskette, RX50 diskette, RL01 or RL02 cartridge, or 1600 bpi magnetic tape) containing at least the following files:

AUTCVT.SAV	Program to convert accounting files to new (v4) format.
BLDTSX.COM	Command file used for "short-cut" installation.
CCL.SAV	SAV file of CCL command processor.
DTSUB.MAC	Subroutines to perform record locking for DIBOL.
FILTIM.SAV	Program to obtain file creation time.
FTSUB.MAC	Subroutines to access RTSORT from FORTRAN.
LOGON.SAV	SAV file of TSX-Plus logon program.
MAKTSX.COM	Command file used for "short-cut" installation.
SETSIZ.COM	Command file to set memory size of system programs.
SETSIZ.SAV	Program to store memory size info. into SAV files.
SYSMON.OBJ	Dynamic system status display program object module.
SYSODT.REL	Program used by system developers to debug TSX-Plus.
TSAUTH.SAV	TSX-Plus account management program.
TSGEN.MAC	Macro source file of TSX-Plus parameter module.
TSXDB.SAV	Program used by system developers to debug TSX-Plus.
TSXLNK.COM	Command file used to link TSX-Plus.
TSXPM.SAV	TSX-Plus performance monitor reporting program.
TSXUCL.SAV	Program to process user-defined commands.

6. Device handler related files:

   a) The following device handlers, which, if necessary, have already been patched and are ready for use: CR, CT, DD, DL, DM, DP, DS, DT, DU, DX, DY, LP, LS, MM, MS, MT, NL, PC, RF, RK, VM, XL; all with the extension .TSX. Note that the VM provided is not the DEC VM handler.

INTRODUCTION

The purpose of this guide is to help you install TSX-Plus on your system in the quickest and easiest way possible. This guide describes the contents of a typical distribution kit, necessary system resources, the circumstances that require a TSX-Plus system generation, the short-cut procedure for installing a functional version of TSX-Plus, and information about how to perform a full system generation to customize TSX-Plus to your needs.

The new TSX-Plus user should follow the short-cut path described in Chapter 2 to verify that all the components of the system are working together correctly and to obtain some experience with TSX-Plus before tailoring TSX-Plus to the peculiarities and needs of a particular site. The experienced TSX-Plus user may proceed directly to Chapter 3 for detailed instructions on performing the TSX-Plus system generation.

This guide is intended for the person installing TSX-Plus and assumes some familiarity with the commands and utilities of RT-11, including: handling of magnetic media; use of command files; the COPY and PRINT commands; use of one of the system editors (KED, EDIT, or TECO); assembling and linking a pre-written MACRO program; and the R[UN] command. It is not necessary to be able to write MACRO programs, but you should be familiar with the assembly and linking process. If you elect to perform a TSX-Plus system generation, it is also useful to be familiar with the usage of comments, labels and parameters in MACRO programs.

It is necessary to have access to configuration information about the various peripheral devices attached to your computer, especially to know what device handlers are needed and what vectors and addresses are used by these devices and by time-sharing terminals. This configuration information should have been written down and left with the system by the person who installed the hardware. If it is not available, ask the person who installed the hardware to enter the configuration information for your system in the form in Appendix B of this manual.

The TSX-Plus operating system is designed to provide the familiar RT-11 operating environment to multiple users. This goal mandates several extensions to the command language of RT-11, and makes a few features unavailable. The TSX-Plus Reference Manual describes all of the functional differences between RT-11 and TSX-Plus, including additional keyboard commands and other features, unsupported commands and other features, and other minor variations between the two operating systems.

The TSX-Plus manuals are intended to be used in conjunction with the RT-11 manuals; refer to the RT-11 manuals for descriptions of features which are not discussed in the TSX-Plus Reference Manual.

1. GETTING STARTED

1.1 Distribution Kit

The TSX-Plus distribution package you have received should contain the following items:

1.* TSX-Plus Reference Manual which describes the features of TSX-Plus.

2.* TSX-Plus System Manager's Guide which provides information needed by the system administrator.

3.* TSX-Plus installation guide.

* Note that only new orders include these three manuals. They are not automatically included with updates.

4. TSX-Plus release notes

5. A magnetic medium (reversible RX01 diskette, RX50 diskette, RL01 or RL02 cartridge, or 1600 bpi magnetic tape) containing at least the following files:

AUTCVT.SAV	Program to convert accounting files to new (v4) format.
BLDTSX.COM	Command file used for "short-cut" installation.
CCL.SAV	SAV file of CCL command processor.
DTSUB.MAC	Subroutines to perform record locking for DIBOL.
FILTIM.SAV	Program to obtain file creation time.
FTSUB.MAC	Subroutines to access RTSORT from FORTRAN.
LOGON.SAV	SAV file of TSX-Plus logon program.
MAKTSX.COM	Command file used for "short-cut" installation.
SETSIZ.COM	Command file to set memory size of system programs.
SETSIZ.SAV	Program to store memory size info. into SAV files.
SYSMON.OBJ	Dynamic system status display program object module.
SYSODT.REL	Program used by system developers to debug TSX-Plus.
TSAUTH.SAV	TSX-Plus account management program.
TSGEN.MAC	Macro source file of TSX-Plus parameter module.
TSXDB.SAV	Program used by system developers to debug TSX-Plus.
TSXLNK.COM	Command file used to link TSX-Plus.
TSXPM.SAV	TSX-Plus performance monitor reporting program.
TSXUCL.SAV	Program to process user-defined commands.

6. Device handler related files:

   a) The following device handlers, which, if necessary, have already been patched and are ready for use: CR, CT, DD, DL, DM, DP, DS, DT, DU, DX, DY, LP, LS, MM, MS, MT, NL, PC, RF, RK, VM, XL; all with the extension .TSX. Note that the VM provided is not the DEC VM handler.

b) Source language patch files are provided for the rare situations in which it is necessary to rebuild the distributed device handlers. All of these files have the extension .SLP.

7. The following object modules are used together with your edited and assembled TSGEN by the command file TSXLNK to build the executable programs TSX and TSKMON: TSX1, TSX2, TSTTY, TSEM2, TSEM3, TSPLAS, TSUSR, TSSPOL, TSLOCK, TSMSG, TSRTX, TSCLO, TSDBUG, TSMIO, TSSLE, TSEXC2, TSKM2A, TSKM2B. SYSMON is used to build the SYSMON utility.

1.2. System prerequisites

1.2.1. Hardware

TSX-Plus is implemented on the PDP-11 series of computers manufactured by Digital Equipment Corporation. TSX-Plus uses the memory management hardware available on most DEC computers. Specifically, TSX-Plus may be used on the following computers: LSI-11/23, PDP-11/23-PLUS, LSI-11/73, MICRO/PDP-11, PDP-11/24, PDP-11/34a, PDP-11/44 and PDP-11/60. (TSX-Plus may not be used on LSI-11/2, PDP-11/03 or VAX computers.) Some older machines which support memory management may also use TSX-Plus. All of these computers support memory up to at least 256Kb. TSX-Plus requires at least 128Kb of memory.

On the LSI-11 bus, the PDP-11/23-PLUS, LSI-11/73, and MICRO/PDP-11 support more than 256Kb of memory whereas the LSI-11/23 may support more than 256Kb of memory only if the backplane is configured for 22-bit addressing. The only Digital devices currently supported with 22-bit DMA addressing on the LSI-11 bus are those using the DL handler (RL01 and RL02), the DU handler (RA80, RC25, RD51, and RX50) and the MS handler (TSV05). All other DMA device handlers only support 18-bit addressing on the LSI-11 bus and therefore must use the TSX-Plus I/O mapping facility if the system has more than 256K bytes of installed memory. On the UNIBUS, the PDP-11/24 and PDP-11/44 support more than 256Kb of memory with the necessary hardware.

1.2.2. Software

Although TSX-Plus completely replaces the RT-11 kernel during operation of TSX-Plus, it must be started from the RT-11 Single-Job or Baseline monitor. In addition, the necessary utilities and device handlers are provided with RT-11. For these reasons, each TSX-Plus site must also be properly licensed to run RT-11.

TSX-Plus supports many of the keyboard commands of RT-11 Version 5, but it may be run with either RT-11 Version 4 or 5. Attempts to use features which require RT-11 Version 5 utilities will generate error messages when used with RT-11 Version 4 utilities. Both Logical Subset Disk and Single Line Editor features are separately implemented in the TSX-Plus kernel, they do not use the LD or SL handler, and are available whether TSX-Plus is used with either RT-11 Version 4 or 5.

2. TSX-Plus SHORT-CUT INSTALLATION PROCEDURE

2.1. Do you need to perform a TSX-Plus system generation?

In order to tailor the features of TSX-Plus for your particular installation TSX-Plus is shipped as several program modules. The module named TSGEN.MAC is provided to allow special modification for the needs of your site. It contains all the parameters which are modifiable by TSX-Plus users and allows selective inclusion of various optional features of TSX-Plus. It also allows specification of the peripheral devices and terminals which are to be used by TSX-Plus.

The short-cut method of installing TSX-Plus is intended to get you running with TSX-Plus immediately without having to do a system generation but with a limited configuration. Once you have gotten the basic system running using the short-cut method, you can perform a system generation to customize the system to your needs and configuration.

The short-cut method of installing TSX-Plus assumes certain system configurations which may be more or less suitable for your site. The short-cut installation makes the following assumptions:

1. The system disk is one of: RL01, RL02, RK06, RK07, RA80, RC25, RD51 or RK05.

2. You have a console terminal at the standard vector and address, and additional terminals are connected through DL(V)11 type interfaces. You must perform a TSX-Plus system generation in order to use DZ(V)11, DH11, or DHV11 multiplexers. The vectors and CSR addresses for time-sharing terminal interfaces are assumed to be:

Line	Vector	Address	
1	60	177560	(operator's console)
2	310	176510	
3	320	176520	

3. Your printer will be spooled (see the TSX-Plus Reference Manual for a discussion of printer spooling); will use the LP device handler; is attached through a LP(V)11 type interface; and is not configured at any of the vector/address combinations defined for time-sharing lines (see list above; the default LP CSR=177514 and vector=200).

5. You do not require any of the following optional features:

   Log-on or device access control.
   Generalized or shared file cache.
   Performance monitoring.
   Shared run-time systems.
   Virtual arrays or virtual overlays.

   (The functions of these features are described in both the TSX-Plus Reference Manual and the TSX-Plus System Manager's Guide.)

6. You do not have or plan to use memory above 256Kb.

If your requirements differ from these, then it will be necessary for you to perform a complete TSX-Plus system generation procedure as described in Chapter 3. Note that the short-cut installation procedure does not copy from the distribution medium all of the files necessary for a complete system generation; if you need to perform a TSX-Plus system generation, copy all of the files from the distribution disk(s) to a working surface.

In cases in which there are no hardware conflicts with the above restrictions, it may be useful to use the short-cut installation method to verify the correct interaction of TSX-Plus with your hardware and to gain experience with the TSX-Plus system before selecting the parameters which must be specified in TSGEN.MAC for generation of the TSX-Plus system.

2.2. TSX-Plus short-cut installation procedure

There are three simple steps to the short-cut installation procedure:

1) Assign the logical device name "IN" to the device containing the TSX-Plus distribution files. If your distribution was supplied on floppy diskette then files must be transferred from both sides of the diskette. The short-cut installation command file (MAKTSX.COM) will prompt you to replace the input volume (turn the diskette over) at the appropriate time. In the following examples, the unit number "n" refers to the unit containing the distribution medium.

If TSX-Plus was distributed on a floppy diskette:

```
.ASSIGN DYn IN
```

or

```
.ASSIGN DXn IN
```

If TSX-Plus was distributed on an RL01 or RL02 disk:

```
.ASSIGN DLn IN
```

If TSX-Plus was distributed on magnetic tape:

```
.ASSIGN MTn IN
```

or

```
.ASSIGN MSn IN
```

or

```
.ASSIGN MMn IN
```

2) Assign the logical device name "DK" to a working surface on which TSX-Plus will be assembled and linked. Approximately 1400 free blocks are necessary on the working surface; about 700 blocks for files copied from the distribution and about 700 blocks for files created during the installation process. For example:

```
.ASSIGN DL1 DK
```

The system disk also requires approximately 2000 free blocks to contain the executable TSX-Plus program, utilities, device handlers, job swapping and spool files.

These figures reflect the minimum total free space required. If insufficient free space is available, remove unnecessary files. If the free space is badly fragmented, it may be necessary to squeeze the fragmented surface.

3) Use the command file provided to copy, assemble, link and start TSX-Plus.

```
.@IN:MAKTSX
```

In order to install TSX-Plus, you must be booted from the RT-11 Version 4, 5 or 5.1 Single-Job or Baseline monitor. The Foreground/Background and XM monitors do not leave enough room to run the TSX.SAV module and will report an error if you try to start TSX-Plus from them. (This is also true of some RT-11 Single-Job monitors which have been "sysgenned" and include too many features, such as multi-terminal support; in this case, copy the original Single-Job monitor (RT11SJ.SYS) from your RT-11 distribution and boot it before running TSX-Plus.) You may also need to SET USR SWAP if the USR has previously been set to NOSWAP.

To reiterate the short-cut installation procedure:

1. Boot the RT-11 Single-Job or Baseline monitor. If you normally use either the Foreground/Background or XM monitor, type the following command:

   .BOOT RT11SJ

2. Assign logical names to the device holding the TSX-Plus distribution files and to the device on which TSX-Plus will be built. For example:

   .ASSIGN DX0 IN
   .ASSIGN DL1 DK

   (DX and DL are examples, use the correct devices for your site.)

3. Prepare at least 2000 free blocks on the system disk, and 1400 free blocks on the work disk. You can verify the presence of the necessary free space with the DIRECTORY commands:

   .DIR/FREE SY:
   .DIR/FREE DK:

4. Execute the command file (MAKTSX.COM) on the distribution medium which copies the necessary files to the system disk and to the working disk, builds, copies to the system disk and starts TSX-Plus.

   .@IN:MAKTSX

   This command file will prompt you to replace the input volume at the appropriate time. If you received TSX-Plus on a floppy diskette, then turn the diskette over at this time and type Y and then press the RETURN key. If you received TSX-Plus on a large disk or on magnetic tape, just type Y and enter a RETURN in response to this prompt.

   In either case, when you are asked to replace the output volume, just type Y and press RETURN.

5. After a few minutes, you should be running TSX-Plus. At this point, a greeting message containing S&H's copyright notice and the date and time should appear on the operator's console. TSX-Plus is now ready to accept a command. The monitor prompt is a period (".").

   The greeting message will appear on other terminals when they are on-line and a carriage return is typed on their keyboards.

   If this short-cut installation does not succeed or if any errors are reported at any point in the process: review the requirements listed

above for the short-cut installation process; review the common errors described in Chapter 6. Chapter 3 provides information about how to do a full system generation.

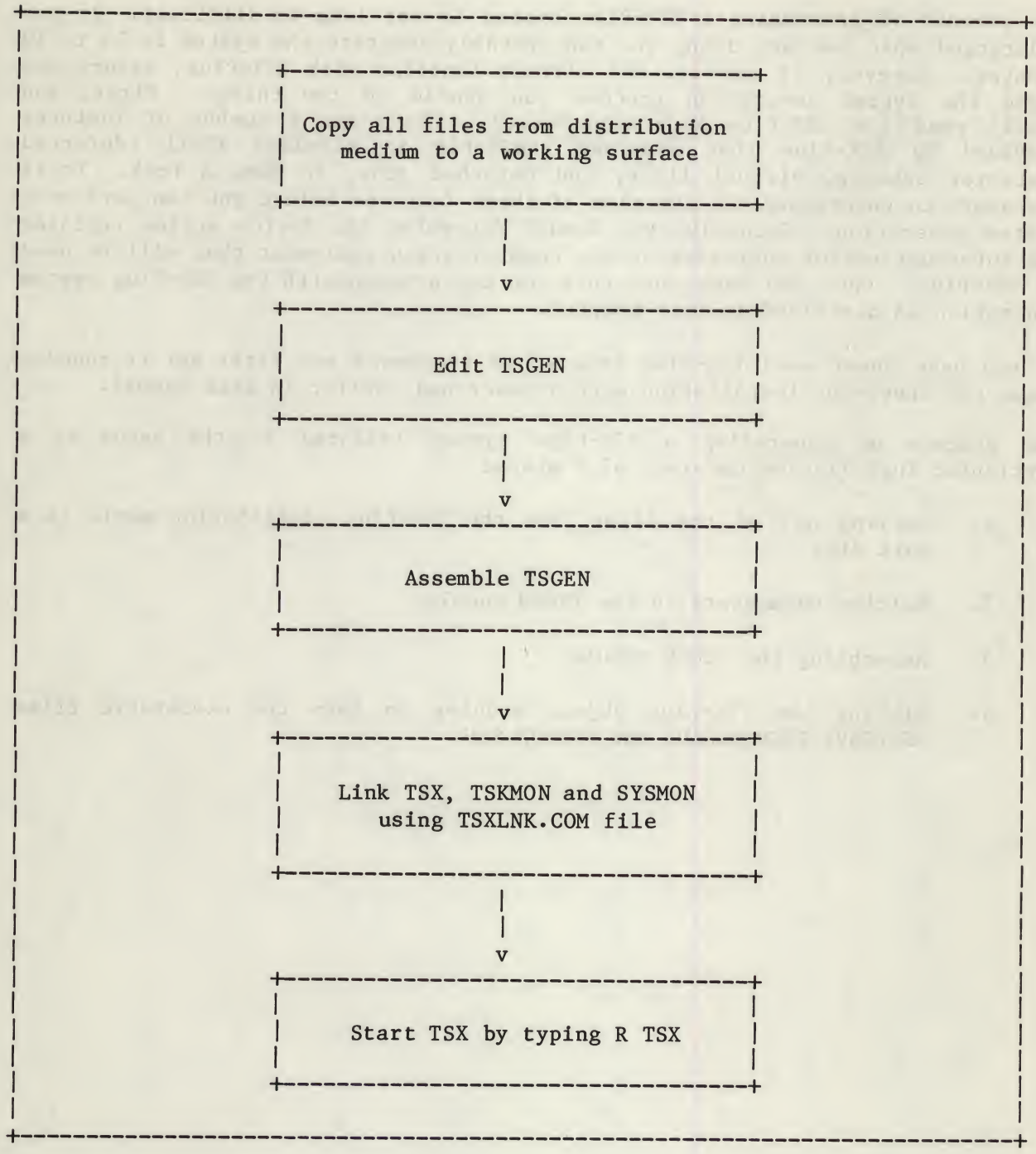
6. Once TSX-Plus is successfully installed, you may subsequently start it, after booting the RT-11 Single-Job monitor, with the command:

```
.R TSX
```

7. You can verify the operation of TSX-Plus at any time while it is running by using the SYSTAT command to display the status of the system and all current time-sharing jobs. The SYSTAT command is described in the TSX-Plus Reference Manual, as are all other valid TSX-Plus commands. The SYSMON utility program may also be used to dynamically display system and job parameters. Use of the SYSMON program is described in the TSX-Plus System Manager's Guide. The SYSMON program may only be used with VT200, VT100 or VT52 type terminals.

3. TSX-Plus SYSTEM GENERATION

3.1 System Generation Overview



The process of generating a TSX-Plus system is not long or difficult. If you understand what you are doing you can probably generate the system in 15 to 30 minutes. However, if you are not already familiar with TSX-Plus, before you begin the system generation process you should do two things. First, you should read the TSX-Plus Reference Manual. There are a number of features provided by TSX-Plus that are not available in standard RT-11 (deferred character echoing, virtual lines, and detached jobs, to name a few). It is necessary to understand the function of these features before you can perform a system generation. Secondly, you should determine the device status register and interrupt vector addresses of the communication equipment that will be used by TSX-Plus. Once you have done this you can proceed with the TSX-Plus system generation as described in this chapter.

If you have never used TSX-Plus before, we recommend you first get it running using the short-cut installation method described earlier in this manual.

The process of generating a TSX-Plus system tailored to the needs of a particular installation consists of 4 steps:

1. Copying all of the files from the TSX-Plus distribution media to a work disk.

2. Editing parameters in the TSGEN module.

3. Assembling the TSGEN module.

4. Linking the TSX-Plus object modules to form the executable files TSX.SAV, TSKMON.SAV, and SYSMON.SAV.

3.2 Editing the TSGEN module

The TSGEN module of TSX-Plus is supplied in source form. TSGEN contains no executable code, but rather contains the definitions of parameters and tables that are used by TSX-Plus. In building a TSX-Plus system, the RT-11 KED, K52, TECO, or EDIT editor program is used to set appropriate values for parameters in TSGEN. This module is then assembled and linked with the other TSX-Plus object modules. Each of the parameters found in TSGEN is described below.

Note that:

1. Numeric values are assumed to be octal unless the number is terminated with a decimal point.

2. When using editors that recognize them, the TSGEN module is divided into several "pages" by form-feed characters.

3. On each parameter needing a file name, the file name is specified as a RAD50 string in the format <DevFilnamExt> with no punctuation (i.e. no colons or periods) and with spaces where there is no character. For example SY:A.TSX would be specified as <SY A TSX>. Note that this does not hold true for the DETACH parameter.

3.3 Setting parameters in TSGEN

Save a copy of the original TSGEN.MAC file, then use an editor to set the appropriate parameter values for the system being generated. The beginning of the parameter section of TSGEN.MAC can be easily located by searching for a string of three equal signs.

3.3.1 General parameters

Parameter	Meaning
SWDBLK	This is the name of the file that will be used to hold programs swapped out of memory by TSX-Plus. The default name is "SY:TSXSWP.TSX". The first three characters of the file specification may be changed to direct the swap file to some other device. The size of the job swap file is determined by the number of time-sharing lines and the amount of memory each may use.
SPLBLK	This is the name of the file that holds output directed to spooled devices. The file name must be supplied even if there are no spooled devices. The default file name is "SY:TSXSPL.TSX". Note that it is possible to place the swap and spool files on separate devices. The size of the spool file is determined by the SPOOL macro (see below).
RSFBLK	This is the name of the PLAS (Program's Logical Address Space) region swap file. This file is used to store memory regions obtained by PLAS when they are swapped out of memory. The default name is "SY:TSXRSF.TSX". The first three characters of the file name may be changed to direct the PLAS region swap file to some other device. PLAS regions are used by programs that have virtual overlays or virtual arrays. The size of the PLAS region swap file is specified with the SEGBLK parameter (see below).
HIMEM	This parameter is used to specify the maximum amount of memory that can be used by any job (exclusive of PLAS regions, such as virtual arrays and virtual overlays). The value is specified in terms of k-bytes. The maximum value that may be specified is 64 (Kb). The value of this parameter does not affect the size of the generated TSX-Plus system; however, it does affect the size of the TSX-Plus swap file whose size is approximately: File size (blocks) = (Total lines) * (HIMEM+4) * 2

DFLMEM This parameter specifies the default memory size to be allocated to a job when it logs on. Specify the value as number of k-bytes. After a line is logged on, the "MEMORY" command may be used to alter the number of kilobytes of memory allocated to the job. The value for this parameter must not be greater than the value for the HIMEM parameter.

SEGBLK This parameter specifies the number of 512-byte blocks to allocate for the swap file that is used for extended memory PLAS (Program's Logical Address Space) regions. These regions are used by programs that have virtual overlays or virtual arrays. The name of the PLAS region swap file is specified with the RSFBLK parameter. If the system is generated as a non-swapping system (SWAPFL=0) then PLAS regions must all fit in memory, no region swap file is allocated or used but the SEGBLK parameter must be set to a non-zero value to cause code to support the PLAS facility to be loaded with the system. Note that this parameter specifies the total space in the PLAS swap file for all extended memory regions in use at any time by all jobs. For example, if a system is to support a maximum of 4 jobs each of which may use 50 Kb of PLAS regions, the total space required is 200 Kb (4 * 50) which requires 400 blocks (1 Kb = 2 blocks) in the region swap file. Actually the file should be allocated with more space than this, since free space in the file may become fragmented as regions are allocated and deallocated. Setting SEGBLK to 0 (zero) disables use of PLAS.

SWAPFL This parameter controls whether TSX-Plus is allowed to swap jobs to disk if insufficient memory is available to hold all active users. The normal case (SWAPFL=1) allows TSX-Plus to do job swapping. SWAPFL can be set to 0 (zero) in special situations such as when a small number of lines are being supported on a floppy disk based system that does not have room for a swap file. If SWAPFL is set to zero the following actions occur:

1. No disk swap file is created.
2. A line will not be allowed to log on if there is insufficient free memory space to support it.
3. Each job is allocated a memory size equal to DFLMEM (default job memory size).
4. Neither the MEMORY command nor EMTs to change the job size can be used.
5. Extended-memory PLAS regions can only be created if there is adequate contiguous free space in memory for them. No PLAS swap file is created.

BUSTYP This parameter defines the machine bus structure for TSX-Plus. There are two possible machine bus structures supported by TSX-Plus - the QBUS (LSI) and the UNIBUS. Select QBUS for 11/23, 11/23-Plus and 11/73, and UNIBUS for 11/24, 11/34a, 11/44 and 11/60.

EXTMCH This parameter, when set equal to 1, enables 22-bit addressing for the 11/23-PLUS, 11/73, 11/24, and 11/44 model CPU's. This should be used when the machine has more than 256 Kb of memory installed. This feature requires the use of 22-bit extended memory mapping for the QBUS or the UNIBUS. If your system (CPU, backplane, and memory) does not support 22-bit addressing, then set this parameter to 0 (zero). It is possible with TSX-Plus to use Q-BUS DMA device controllers which only support 18-bit addressing, however because this is supported by transferring the data through an intermediate system buffer this type of I/O suffers some speed degradation. For this reason it is strongly recommended that on Q-BUS systems the system device controller and handler support 22-bit addressing if the system is to be used with more than 256 Kb of memory (EXTMCH = 1). See the description of the MAPIO modifier to the DEVDEF macro for further information on system mapping of I/O transfers. The only Q-BUS DMA device handlers supported by Digital for 22-bit addressing are DL, DU and MS.

MEMSIZ This parameter controls the maximum memory available for TSX-Plus system use. The value is the memory upper limit size specification expressed in number of k-bytes. Memory above this upper limit will not be used by the operating system. If the MEMSIZ parameter is set to 0 (zero), TSX-Plus will use all available memory on the machine. To disable the use of extended memory, set MEMSIZ to 248 or less (but greater than zero). On machines with a large amount of memory, it is convenient to set an upper limit on the amount of memory to by used by TSX-Plus so that the virtual memory handler (VM) may use the remainder as a RAM based pseudo disk device. This is especially useful for compiler intermediate files. See the section on the VM handler for more information on use of VM with TSX-Plus.

INIABT This parameter controls the action taken by TSX-Plus when certain errors are detected during system initialization. If INIABT is set to zero, TSX-Plus ignores the error and continues running. If INIABT is set to one, TSX-Plus aborts the initialization and prints an error message. The following initialization errors are controlled by the INIABT flag:

1. A device that was specified in TSGEN is not installed in RT-11 or does not have a TSX handler on the system disk.
2. A time sharing line that was generated into TSX-Plus is not installed on the machine.
3. A shared run-time system file could not be found during startup.

IOABT This parameter controls the action taken by TSX-Plus when a job terminates execution. If IOABT is set to zero, TSX-Plus will wait for all outstanding I/O pending for the job to complete before the job is actually terminated. If IOABT is set to one, TSX-Plus will call the handler abort entry point for all outstanding I/O pending for the job. This parameter is usually set to 1 (one). The SET IO

command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET IO command. The IOABT parameter must be set to one if the XL or CL handlers are used with VTCOM.

U$CL This parameter controls whether or not support for user-defined keyboard commands is included in the system. If U$CL is non-zero, TSX-Plus calls on the TSXUCL program to process user-defined commands. If U$CL is zero, user defined commands are not supported by the system. Note that if U$CL is set non-zero, the TSXUCL.SAV file should be on the system disk when TSX-Plus is started.

UCLMNC This parameter sets the maximum number of user-defined commands that may be declared by each job. It also determines the size of the file used to store these definitions (SY:TSXUCL.TSX). The size of this file, in blocks, is approximately:

File size (blocks) = UCLMNC * (Total lines) / 5

UCLORD This parameter specifies the default order in which the TSX-Plus command interpreter checks for user-defined commands. The UCLORD parameter should be equated to one of the symbolic names FIRST, MIDDLE, LAST, or NONE. The SET UCL command may be used to change the order of command interpretation for a job. See the description of keyboard command interpretation in the TSX-Plus Reference Manual for a full discussion of command processing order.

LDSYS This parameter controls whether the standard system support for logical disks is to be included in the system. If LDSYS=1, system support is provided for logical disks; if LDSYS=0, system support is not provided for logical disks. Normally logical disk support should be included; however system support for logical disks may be excluded if a specialized LD handler providing custom logical disk support is being used rather than the standard system support.

SLEDIT This parameter controls whether support for the Single Line Editor facility is included in the system. If SLEDIT=1, the single line editor is included in the generated system; if SLEDIT=0, the single line editor is not included in the system. Use of the single line editor for a given time-sharing job is controlled by use of the SET SL command; however the SLEDIT parameter must be set to 1 (one) if the single line editor facility is to be made available to any lines.

DBGFLG This parameter controls whether support for the program debugging facility is included in the system. If DBGFLG=1, the debugging facility is included in the generated system; if DBGFLG=0, the debugging facility is not included in the system. See the TSX-Plus Reference Manual for information about the use of the program debugging facility.

QUAN0 This parameter specifies the time-slice value used to schedule jobs with user-specified priorities equal to or greater than the PRIHI parameter. High priority jobs that have the same priority are scheduled on a round-robin basis using QUAN0 as the time-slice value. If QUAN0 is set to 0 (zero), high-priority jobs are not time-sliced. Specify the value of QUAN0 in 0.1 second units. The SET QUAN0 keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information about the SET QUAN0 command.

QUAN1 This parameter specifies the length of time a job will run in an interactive state after receiving input from the terminal. Specify the value in 0.1 second units. A job is classified as "interactive" and given a priority boost each time it receives input from the terminal. If the job uses up more than QUAN1 units of CPU time before it receives more input from the terminal, the job is classified as "non-interactive" and runs at normal priority. The SET QUAN1 keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN1 command.

QUAN1A This parameter specifies the length of time a non-interactive job will run in a high-priority state after being restarted from a wait state. Increasing the value of this parameter tends to give priority to I/O active jobs and allow them to dominate over other jobs. The SET QUAN1A keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN1A command.

QUAN1B This parameter specifies the execution time-slice value for round-robin scheduling of interactive jobs. Specify the value in 0.1 second units. The SET QUAN1B keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN1B command.

QUAN1C This parameter specifies the length of time a job will execute in the highest priority interactive state after receiving an activation character. Specify the value in 0.1 second units. The SET QUAN1C keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN1C command.

QUAN2 This is the time-slice given to compute-bound jobs. A compute-bound job is allowed to run this long if there are no high-priority tasks that need service. Specify the value in 0.1 second units. The SET QUAN2 keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN2 command.

QUAN3 This is the time-slice used for round-robin scheduling of jobs with user-assigned priority values less than or equal to the PRILOW parameter. Specify the value in 0.1 second units. The SET QUAN3 keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET QUAN3 command.

INTIOC This parameter controls the scheduling of interactive jobs which also do non-terminal I/O. An interactive job which exceeds this number of I/O operations before receiving another activation character will be rescheduled as a non-interactive job. This parameter should be large enough to keep jobs that are operator intensive in an interactive state. The SET INTIOC keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET INTIOC command.

HIPRCT This parameter controls the scheduling of non-interactive jobs which do non-terminal I/O. On completion of non-terminal I/O, jobs are usually scheduled into a high-priority state. However, a job which exceeds this number of I/O operations will be rescheduled in the normal priority compute-bound state. The SET HIPRCT keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET HIPRCT command.

CORTIM Each time a job is swapped into memory from disk a timer is started for that job. The job is not eligible to be swapped out of memory until CORTIM units of time have elapsed. However, a job becomes immediately eligible to be swapped if it goes into any wait state other than non-terminal I/O, regardless of the value of CORTIM. Specify the CORTIM parameter value in 0.1 second units. The SET CORTIM keyboard command may be used to dynamically alter this parameter during system operation. See the TSX-Plus Reference Manual for more information on the SET CORTIM command.

PRILOW This parameter specifies the highest user-specified job priority that is part of the fixed-low-priority group. Jobs with priorities less than or equal to PRILOW are considered low priority jobs and execute at fixed priorities below normal time-sharing jobs. The value of PRILOW must be in the range 0 to 126, and must be less than PRIHI.

PRIHI This parameter specifies the lowest user-specified job priority that is part of the fixed-high-priority group. Jobs with priorities greater than or equal to PRIHI are considered high priority jobs and take precedence over normal time-sharing jobs. Priorities in the fixed-high-priority group are normally reserved for real-time jobs and should never be assigned to normal time-sharing jobs. The value of PRIHI must be in the range 1 to 127, and must be greater than the value specified for PRILOW.

PRIDEF This parameter specifies the default job priority that will be assigned to jobs. The SET PRIORITY keyboard command may be used to dynamically set job priority, and priority may also be set from within a job by use of an EMT. The value of PRIDEF must be in the range 0 to 127. Normally, PRIDEF should be greater than PRILOW and less than PRIHI.

PRIVIR When a job switches to a virtual line, the job execution priority of the disconnected line is reduced by this amount. This automatic priority reduction does not apply to jobs with priority values less than or equal to PRILOW or greater than or equal to PRIHI. Also, jobs with priorities in the normal time-sharing range, between PRILOW and PRIHI, will never have their priority reduced to less than (PRILOW+1). See the TSX-Plus System Manager's Guide for more information on priority and job scheduling.

MAXSEC This parameter is used to specify the maximum number of virtual lines that a single user may own at any given time.

MAXFIL Maximum file size (number of blocks) that will be returned in response to a .ENTER programmed request that specifies a file size of 0 blocks. This parameter does not limit the space that will be allocated to .ENTER requests that specify a size. Rather, it only affects .ENTER requests that specify a file size of 0. If a value of 0 (zero) is specified for MAXFIL, no limit is placed on the size of a file created with a specified size of 0.

CACHE This parameter controls the number of 512 byte data blocks allocated in extended memory for use by the generalized data caching facility. Data caching is a technique for improving system performance by keeping in memory a "cache" of the most recently accessed blocks of data. Use of the generalized data cache is not recommended for systems with less than 256Kb of memory. If generalized data caching is not wanted, set CACHE=0. If generalized data caching is wanted, set CACHE to the number of 512 byte blocks to be allocated for the cache.

In selecting this parameter, consideration must be given to the tradeoff between the improvement to system performance to be gained by data caching versus the decrease in total free memory space for jobs which may cause increased job swapping. While data cache buffers are not included in the low memory area, they do remove space from that available to user jobs. Generally it is recommended that CACHE be set to zero if less than 256Kb of memory is installed on the system or if the system is primarily bound by CPU utilization rather than I/O throughput. If data caching is used at all, it is recommended that CACHE be set to at least 50.

One way to determine the best value for this parameter is to generate a system with a large number of cache buffers and then use the SET

CACHE keyboard command to vary the number of buffers used while observing the effect on system performance.

MAXCSH The MAXCSH and NMFCSH parameters relate to the cache of file directory entries maintained by TSX-Plus. This cache is used to reduce the number of disk accesses required to do .LOOKUPs on frequently accessed files. The system disk directory is always cached. Other devices are only cached if they are introduced to the system by use of the "MOUNT" command. File directory caching can have a dramatic affect on the speed of .LOOKUPs of commonly used files. It does not affect the time taken to do .ENTER, .DELETE and .RENAME requests. The MAXCSH parameter is used to specify the maximum number of device units whose directories may be cached. Note: the value of MAXCSH must be large enough to include all mounted logical disks as well as mounted physical devices.

NMFCSH This parameter specifies the maximum number of file entries that can be held in the file directory cache. This number is the total number of file entries that will be cached for all users on the system (the cache is common to all users).

MAXALC This parameter specifies the maximum number of device units that can be allocated to jobs for exclusive use by use of the ALLOCATE command.

MAXMON This parameter controls the maximum number of job monitoring requests which can be simultaneously active system-wide. The TSX-Plus job monitoring facility allows one job to schedule a completion routine to be entered with a status code whenever the status of a job being monitored changes. See the TSX-Plus Reference Manual for further information about the job monitoring facility. If the job monitoring facility is not wanted, set MAXMON to 0 (zero).

TIMOUT This parameter is only used for lines connected to dial-up telephone equipment. It specifies the length of time that the carrier signal may be lost before the system assumes the connection has been broken and logs off the job. Specify in 0.5 second units.

Setting this parameter to a reasonable value (say one minute) provides time for a disconnected user to reestablish the connection by recalling the computer. You should not make this parameter too large because it provides a window during which another user could dial into the computer and connect to a line that is still logged on for a user who has lost the phone connection.

OFFTIM This parameter is only used for lines connected to dial-up telephone equipment. It is the length of time that a user may remain connected to a dial-up line before logging in. It also controls the length of time that a user may remain connected to a dial-up line between logging off and logging back on. If the user does not log in within

this time interval, the Data Terminal Ready (DTR) signal will be dropped causing the phone to be hung up. Specify this parameter in 0.5 second units.

TSLICH This is the "lead-in" character that tells TSX-Plus that the following character, which is being output by the program, is to be interpreted by TSX-Plus as a special command (for example, defining a new activation character). See Chapter 6 of the TSX-Plus Reference Manual for more information on program controlled terminal options. The default value for this parameter is 35 (29 decimal).

VLSWCH This is the character used to signal a request to switch to a virtual line. See Chapter 3 of the TSX-Plus Reference Manual for a discussion of the use of virtual lines. The default value is 27, control-W.

MXSPAC TSX-Plus allows running programs to dynamically define activation characters. (An activation character is a character which completes a terminal input field, such as carriage return.) MXSPAC specifies the maximum number of user defined activation characters that each line may define.

EDITOR This parameter specifies the default system editor. The SET EDIT keyboard command can be used to select a different default editor for a job. The allowable editors are EDIT, TECO, KED and K52.

WILDFL This flag sets the system default for implicit or explicit wildcards in file names. The SET WILDCARDS command can be used to alter this setting for a line. Specify 0 (zero) for explicit wildcards or 1 for implicit wildcards. See the RT-11 System User's Guide for further discussion of explicit and implicit wildcards.

3.3.2 Device definitions

The next section of TSGEN is used to specify all of the devices which are to be available while TSX-Plus is running. Unlike RT-11, TSX-Plus does not allow fetchable, non-resident device handlers. All TSX-Plus device handlers are loaded into memory when the system is started and remain in memory until the system is stopped. Some TSX-Plus device handlers must be loaded into the 40Kb low-memory system area, but other handlers can be loaded into extended memory regions.

Three macros are used to specify devices to TSX-Plus: the DEVBEG macro begins the device specification section; the DEVEND macro ends the device specification section. The DEVDEF macro is used to specify the name and options for each device. Thus, the DEVDEF macro is used once for each device between the DEVBEG and DEVEND macros.

The following devices are an integral part of the TSX-Plus system and should not be specified with the DEVDEF macro: CL, LD, TT, and SL. It is recommended that the CL handler be used to drive all serial devices instead of LS and XL. See the TSX-Plus System Manager's Guide for a description of the CL handler.

DEVDEF The DEVDEF macro must be used to define the names and characteristics of all devices which are to be available to TSX-Plus users.

The form of the DEVDEF macro is:

```
DEVDEF <dd>[,option,...,option]
```

The first parameter defines the 2 character device name. The device name must be enclosed in angle brackets ("<" and ">").

The optional parameters specify the device characteristics. There are nine allowable device attributes which may be specified in any order. They are as follows:

DMA Device performs Direct Memory Access (DMA). In UNIBUS systems with more than 256K bytes of memory, TSX-Plus allocates and controls Unibus Mapping Registers (UMR's) to perform I/O requests for a DMA device.

MAPIO Perform I/O mapping. This parameter should be specified only for Q-BUS DMA devices which have 18-bit controllers or handlers (such as RX02 -- DY) but are being used in an otherwise 22-bit environment. MAPIO should not be specified for any NONDMA devices nor for any DMA device for which the handler and controller actually support 22-bit addressing. MAPIO should likewise never be specified for any device in an 18-bit environment (EXTMCH = 0; 256Kb or less memory; 11/23 without modified backplane). Older LSI-11/23 systems can be usually be upgraded to support 22-bit memory by changing to the H9275-A backplane. In

addition, 18-bit DMA controllers should be replaced with 22-bit controllers whenever possible if more than 256 Kb of memory will be used. MAPIO should never be specified for any device in a UNIBUS system.

EVNBUF Require even byte buffer address for I/O transfers. Some device controllers (especially DMA devices) and device handlers (VM) which implement a word transfer (rather than byte), require the buffer address to begin on a even byte address (word aligned). In these cases, odd byte addresses may cause I/O failure or fatal system errors which could halt the machine's execution. If EVNBUF is specified, TSX-Plus will check the buffer address to insure that the transfer is word aligned. If the I/O request does not begin on a word boundary, a user error will be returned from the EMT request.

NOCACHE Do not use generalized data cache for this device. For certain devices, it is desirable to disable generalized data cache. For example, since the VM handler uses memory as a device, it would be wasteful of machine resources to also allow it to utilize generalized data cache. This would not only result in displacement of information contained within the cache but would also have the additional overhead of a useless memory to memory transfer.

NOMOUNT Do not allow mounts for this device.

REQALC Require device allocation before use. If this option is specified, access to the device units is only allowed to users who have first allocated the device by use of the ALLOCATE command.

MAPH Load the device handler outside the low memory 40K byte region and into a mapped handler region. TSX-Plus will place device handlers within an extended memory region, reducing the size of the low memory kernel region (restricted to 40K bytes). Handlers which are placed in extended memory are known as "mapped" handlers. TSX-Plus communicates with mapped device handlers by mapping PAR 5 to the handler's extended memory base address. As device handlers are loaded, the interrupt entry point is intercepted and directed to a low memory routine which will map to the handler then enter the handler's interrupt entry code.

Handlers may be mapped under the following conditions:

1. Since only one PAR register is used to access the device handler it must not be larger than 8K bytes.

2. Since handlers are accessed by kernel PAR 5, the handler must not use kernel PAR 5.

3. Since only two device interrupt vectors per handler are redirected, the handler may not connect to more than two device interrupt vectors. In addition, since the redirection is performed only once, during initialization, the handler may not dynamically connect to interrupt vectors.

4. When the device handler contains an internal buffer used for DMA access, it must calculate the correct physical address taking into account it's own mapped address. It must also declare the HANBUF option (see below) which will not allow it to be mapped on UNIBUS systems with more than 256Kb of memory or when MAPIO is also specified.

See the System Manager's Guide for further information about mapped handlers.

NOMAPH Do not load the handler into a mapped handler region; instead load it into the low memory 40K byte region. Some device handlers are not eligible for mapping into extended memory regions and TSX-Plus will place them in the low memory kernel region. The NOMAPH option can be used to specify that a handler which would ordinarily be mapped should not be mapped. This option takes precedence over the MAPH option.

HANBUF Handler contains an internal I/O buffer used for DMA transfers. Handlers with internal DMA buffers require special coding to be used as a mapped device handler. In addition, when TSX-Plus is evaluating the system definitions and device characteristics for loading device handlers, it will never map a handler which uses an internal buffer if the handler also requires mapped I/O transfers in QBUS systems with more than 256Kb of memory (MAPIO option), or in UNIBUS systems with more than 256Kb of memory.

The following default options are set by the system for standard devices:

Device	Options
CR	MAPH
CT	MAPH
DD	NOMAPH
DL	DMA,MAPH,HANBUF
DM	DMA,NOMAPH
DP	DMA
DS	DMA
DT	DMA
DU	DMA
DW	MAPH
DX	MAPH
DY	DMA,NOMAPH
DZ	MAPH
FW	DMA
LP	MAPH
LS	MAPH
MM	DMA,MAPH,HANBUF
MS	DMA,MAPH,HANBUF
MT	DMA,MAPH,HANBUF
NL	MAPH
PC	MAPH
RF	DMA
RK	DMA,MAPH
VM	EVNBUF,NOCACHE,NOMAPH
XC	MAPH
XL	MAPH

If neither the MAPH nor the NOMAPH options is specified for a handler, the handler is not mapped. If both the NOMAPH and MAPH options are specified, the handler is not mapped. Thus it is possible to override the default MAPH option for a standard handler by specifying NOMAPH.

The devices CL, TT, LD, and SL do not require device definitions and should not be included in the DEVDEF table. These devices are an integral part of TSX-Plus and do not require separate device handlers. See the section on device handlers in the System Manager's Guide for more information.

A maximum of 15 devices can be installed in TSX-Plus as distributed, including CL, TT and LD. SL is not implemented as a pseudo-device in TSX-Plus.

By convention, the system device (device from which RT-11 was booted and TSX-Plus is run) should be the first device definition.

For example:

```
DEVBEG                  ;Beginning of device definitions
DEVDEF    <DL>
DEVDEF    <RK>,MAPIO
DEVDEF    <DY>,MAPIO
DEVDEF    <MS>
DEVDEF    <LP>
DEVDEF    <NL>
DEVDEF    <VM>
DEVEND                  ;End of device definitions
```

It is not necessary to have device handlers installed in RT-11 when TSX-Plus is started. The only device handlers that must be installed are handlers for the devices where the TSX-Plus swap and spool files are placed.

TSX-Plus will refuse to install new device handlers which were issued by Digital subsequent to the version of RT-11 under which TSX-Plus is being started. In other words you must upgrade to the appropriate version of RT-11 in order to be able to use the newer device drivers with TSX-Plus. Specifically, the following device drivers require the indicated version of RT-11:

Device	RT-11 version
DU	5.00
XL	5.01

MIONBF This parameter specifies the number of system I/O buffers to be allocated for I/O mapping. I/O mapping is used for devices with 18-bit controllers or handlers being used with 22-bit Q-bus systems. The MAPIO parameter to the DEVDEF macro specifies which devices require I/O mapping. One buffer should be allocated for each device which requires I/O mapping and which will be in use simultaneously with other devices which also require system I/O mapping. For example, if both RK and DY need system I/O mapping, but both devices will never be in use at the same time, then 1 buffer would be adequate. If however, both devices are likely to be in use at the same time, then 2 buffers should be allocated. These buffers are shared by the system and all user jobs that are doing I/O to devices needing system I/O mapping. If a transfer is requested to a device which requires system I/O mapping and a buffer is not available, the transfer will be delayed until a buffer becomes available.

MIOBSZ This parameter specifies the size of the buffers used for system I/O mapping. The value specifies the number of 512 byte areas to allocate for each buffer. The larger this parameter is, the faster system mapped I/O transfers will occur. The maximum value for this parameter is 15. Because directory operations are performed in 1024 byte chunks, if system I/O mapping is selected at all, the minimum recommended number for MIOBSZ is 2. It is strongly recommended that the system device not require I/O mapping. However, if the system device does require I/O mapping, MIOBSZ should be set to 15, the maximum value.

3.3.3 Device spooling parameters

Device spooling is an optional feature of TSX-Plus. If any spooled devices are wanted, give appropriate values to the parameters of the SPOOL macro. If spooling is not wanted, specify 0 (zero) as the first parameter to the SPOOL macro; the other parameters will be ignored. See Chapter 5 of the TSX-Plus Reference Manual for more information on device spooling.

SPOOL The SPOOL macro is used to define information about devices that are to be spooled by TSX-Plus. The form of the SPOOL macro is:

```
SPOOL  ndev,nfile,nbuf,nblocks,<dev...>,hold,nback
```

The meanings of these parameters are:

ndev The number of devices that are to be spooled by TSX-Plus. Specify 0 (zero) if there are none.

nfile The number of spooled files that may be open to all users. A spooled file entry is required for each file that is being printed, waiting to be printed, or is in the process of being generated by a running program for printing on a spooled device.

nbuf The number of 512 byte buffers that are to be used by the spooling system. If two buffers are available for each active device, the I/O will be "double-buffered" to achieve maximum speed. If fewer buffers are available than active devices, the devices will operate in bursts and share the buffers. Space for these buffers is allocated in the mapped portion of TSX-Plus.

nblocks The number of disk blocks to be allocated within the spool disk file. All spooled files share this space in the common disk spool file. If the file fills up, running programs are suspended until space becomes available as blocks are printed and released.

<dev> The names of those devices that are to be spooled. Specify exactly three characters per name. The spooled devices must be non-file structured output devices such as line printers, plotters, or Communication Lines (CL). Note that spooled devices (except for CL lines) must also be specified in the DEVDEF (device definition) list. See the Device Handlers chapter in the System Manager's Guide for further information about the use of CL lines.

hold Specify 1 for the this parameter if the default mode for the spooler is to be 'HOLD' (see SPOOL command description in the TSX-Plus Reference Manual). Specify 0 (zero) for 'NOHOLD' mode. In HOLD mode, spooled output will not be processed until the spool file is completely created and the I/O channel associated with the file is closed. In NOHOLD mode, a spool file may begin to be copied to the spooled device while the spool file is being created. This mode may be changed dynamically with the SPOOL <dev>,[NO]HOLD command. This mode may also be controlled from within programs on an individual file basis with an EMT request.

nback This parameter specifies the number of spool blocks that TSX-Plus will back up in response to the SPOOL <dev>,BACK command (see SPOOL command description in the TSX-Plus Reference Manual, Chapter 5).

Example:

The following SPOOL macro declares that there are 2 spooled devices: LP and CL2; there may be up to 10 active spooled files; four 512 byte buffers are to be used for spooling I/O; the spool file is to be 500 blocks large; default mode is HOLD; and the SPOOL BACK command is to backup 10 blocks.

```
SPOOL 2,10.,4,500.,<LP CL2>,1,10.
```

3.3.4 Record locking parameters

If the shared file record locking and data caching feature of TSX-Plus is wanted, the three parameters MAXSF, MAXSFC, and MXLBLK must be given appropriate values. If the shared file record locking and data caching facility is not wanted, set MAXSF, MAXSFC, and MXLBLK to 0 (zero).

MAXSF — MAXSF specifies how many shared files may be open simultaneously. Note that several users accessing the same shared file count as one open shared file.

MAXSFC — Maximum number of I/O channels that all users may simultaneously have open to shared files. Note that this is the total number of channels for all users not for each user.

MXLBLK — Maximum number of file blocks that may be simultaneously held locked by any channel. A file block contains 512 characters.

NUMDC — Number of 512-byte data blocks to be allocated for shared file data caching. There are two data caching facilities in TSX-Plus: a generalized data caching facility that caches blocks from all files, and a shared-file data caching facility that only caches blocks from files declared to be "shared" to the system. Both data caching facilities should not be used at the same time. The generalized data caching facility is controlled by the CACHE parameter (see above); the shared-file data caching facility is controlled by the NUMDC parameter.

The shared file data caching facility provides data caching only for files which have been declared as shared files (regardless of access and protection category). Data caching causes the most active blocks for shared files to be held in memory cache buffers. This eliminates all disk I/O when these blocks are read. Data caching is particularly effective for COBOL-Plus ISAM files. The NUMDC parameter controls the number of 512-byte cache buffers that are allocated for data caching. If NUMDC is set to 0 (zero) shared file data caching is not done (but generalized data caching will be done if CACHE is non-zero).

In selecting this parameter, consideration must be given to the tradeoff between the improvement to system performance to be gained by data caching versus the decrease in total free memory space for jobs which may cause increased job swapping. While data cache buffers are not included in the low memory area, they do remove space from that available to user jobs. Generally it is recommended that NUMDC be set to zero if less than 192Kb of memory is installed on the system or if shared files are not accessed heavily. If data caching is used at all, it is recommended that NUMDC be set to at least 5. One way to determine the best value for this parameter is to generate a system with a large number of cache buffers and then use the SET NUMDC keyboard command to vary the number of buffers used while observing the effect on system performance.

3.3.5 Message communication parameters

If the message communication feature is not wanted, the four parameters MAXMC, MSCHRS, MAXMSG, and MAXMRB should be set to zero. If the message communication feature is wanted, assign appropriate values to the four parameters.

MAXMC — Maximum number of message communication channels that may be simultaneously active. A message channel is active if any messages are pending on it or if any users are waiting for messages to come through it.

MSCHRS — Maximum length of messages; specify in bytes.

MAXMSG — Maximum number of messages that may be simultaneously held in message queues for all channels. Note, this is the maximum number of messages that can be queued on all channels, not each channel.

MAXMRB — Maximum number of requests for messages that may be held by the system for all jobs. This includes requests for messages that test for a pending message, wait for a message, or call a completion routine when a message arrives.

3.3.6 Real-time program support parameters

TSX-Plus provides a real-time program support facility that allows multiple real-time programs to be run concurrently with normal time-sharing operations. Note: real-time program support must be included in the system if the SYSMON system monitor display program is to be used.

See Chapter 11 in the TSX-Plus Reference Manual for a full discussion of the TSX-Plus real-time facilities. The basic functions provided by this facility are summarized below.

1. The ability to map the I/O page into the user's virtual memory region so that device status and control registers may be directly accessed by the program.

2. The ability to connect device interrupt vectors to program interrupt service routines. System service support is restricted with this method, but it is quite fast.

3. The ability to connect device interrupt vectors to program completion routines. These real-time completion routines run at user-selectable real-time priority levels that preempt execution of normal time-sharing jobs.

4. The ability for a program to lock itself in memory so that rapid interrupt response can be assured.

5. The ability for a program to dynamically set its execution priority.

6. The ability for a program to suspend its execution until an interrupt occurs.

7. The ability to convert a virtual address within the job's region to a physical address for DMA I/O control.

8. The ability to map a virtual address region to a physical address region.

9. The ability for a program to declare a list of addresses of device control registers to be reset when the program exits or aborts (.DEVICE EMT).

RTVECT The RTVECT parameter controls whether or not the TSX-Plus real-time program support facility will be included in the generated system. If real-time program support is not wanted, set the RTVECT parameter to 0 (zero). If real-time program support is wanted, set RTVECT to the number of real-time interrupt vectors that will be used by all real-time programs. If some of the real-time support features are wanted, but no interrupt vectors are necessary, then set RTVECT to 1.

! If the SYSMON system monitor display program is to be used to monitor system performance, real-time support must be included in the system and RTVECT must be at least 1.

NIET MEER!

3.3.7 Performance monitor parameter

TSX-Plus includes a performance monitor facility that allows you to monitor the execution of an application program running under TSX-Plus and produce a histogram showing the amount of time spent in various regions of the program. See Chapter 13 of the TSX-Plus Reference Manual for information on use of the performance monitor feature.

PMSIZE This parameter specifies the number of bytes of memory to set aside for use in accumulating histogram values during a performance analysis run. Memory space equal to the size specified with PMSIZE is allocated in a mapped data region of TSX-Plus for use by the performance analysis facility. If you do not intend to use the performance analysis feature, set PMSIZE to 0 (zero) to avoid using any memory space for this feature. The maximum value that may be given to PMSIZE is 8192.

3.3.8 Shared run-time systems

TSX-Plus supports shared run-time systems. These are reentrant programs or common data buffers that can be shared by multiple users. The RTDEF macro is used to declare shared run-time systems. The form of this macro is:

```
RTDEF    <program-name>,r-flag,skip-count
```

where "program-name" is the 12 character name of the file containing the run-time system. This must be specified in the form <DevFilnamExt>, that is, three characters for the device name, six characters for the file name and three characters for the extension. "r-flag" is either "R" if user programs are to have read-only access the run-time system, or "RW" if read-write access is to be granted. Most run-time systems will use read-only access. Read-write access is primarily useful when the shared run-time facility is being used to provide common data areas being accessed and updated by multiple jobs. The "skip-count" parameter is the number of blocks to be skipped over at the front of the run-time system file when loading it into memory.

Run-time system files are normally SAV files. However any type of file could potentially be used. TSX-Plus simply reads it into memory (without interpreting its contents) and maps portions of it into the job space as requested by EMT's. Shared run-time systems are loaded into memory below the mapped system overlay regions. See Chapter 12 of the TSX-Plus Reference Manual for information about using a shared run-time system.

Examples of shared run-time declarations:

```
RTDEF    <SY CBR050SHR>,R,1.      ;COBOL-Plus runtime
RTDEF    <SY DBLSHRRTS>,R,1.      ;DBL runtime
RTDEF    <RK2COMDT1SAV>,RW,0.
```

3.3.9 Time-sharing line definitions

TSX-Plus supports communication through serial lines connected to DL(V)11, DZ(V)11, and DH(V)11 controllers. Terminals, modems, printers, computers, and other external devices may be connected to these lines.

Serial ports may be used in two distinctly different ways: either as time-sharing lines or as communication lines. A time-sharing line allows a user to control the system: the user can log on, edit files, run programs, etc. A communication line (CL) on the other hand is merely a serial Input/Output port with no more control capability than any other peripheral device. Communication lines are used to drive printers and other serial devices, and to communicate with other computers or external devices.

When the system is generated, serial ports are designated as being time-sharing lines or communication lines. Communication lines require less space in the system than time-sharing lines but are dedicated for use as CL lines. Lines generated as time-sharing lines are normally used in that fashion but may be taken over while the system is running for use as CL lines.

In this discussion a serial port is referred to as a "line". The control facility within TSX-Plus which drives an individual CL line is referred to as a "unit". Each CL line which is generated into the system has a CL unit initially assigned to it. The CLXTRA parameter (see below) is used to include extra CL units in the system which are initially not assigned to any line but which may be used to take over time-sharing lines for use as CL lines. The total number of CL units is equal to the number of dedicated CL lines plus the value of the CLXTRA parameter.

CL units are accessed as I/O devices with the names "CL0:", "CL1:", ..., "CL7:". It is suggested that dedicated CL lines be used to drive serial devices such as printers and plotters. Extra CL units can be included to allow lines that are connected to modems to be used as normal time-sharing lines part of the time and dial-out communication lines at other times.

The CL facility can be used as a replacement for both the LS and XL handlers. The advantage of CL is that a single copy of the handler can drive up to 8 devices connected to DL11 or multiplexer type lines, and it provides bi-directional transfers.

The system may be generated with any combination of time-sharing and dedicated CL lines provided that there are not more than 8 CL units and the total number of lines does not exceed 31. Time-sharing and CL lines may be mixed on the same multiplexer. See the Device Handlers chapter in the TSX-Plus System Manager's Guide for further information about the use of CL lines.

3.3.9.1 Default values for time-sharing lines. The following parameters establish default values which will be used for all lines unless overridden by parameters specified within line definition blocks.

DINSPC This parameter specifies the default number of characters that will be reserved for the input ring buffer for each time-sharing line. This value is used for all virtual lines and for actual lines that do not have any other value specified. It must be large enough to hold an entire line of input plus any characters that are typed ahead. Input ring buffers are not allocated for CL lines.

DOTSPC This parameter specifies the default number of characters that will be reserved for the output ring buffer for each time-sharing line. This value is used for all virtual lines and for actual lines that do not specify any other value. A running program will be suspended when its output ring buffer is filled. The CLORSZ parameter (see below) specifies the default output ring buffer size for CL lines.

OTRASZ A job's execution is suspended and the job may be swapped out of memory when that job's character output buffer is filled. As the output buffer is emptied the job is reactivated when the number of characters remaining in the buffer equals OTRASZ. The idea is to get the job running again before all of the available output is exhausted.

NCSILO TSX-Plus allocates a character storage area for each time-sharing and CL line to hold characters before they are processed and moved to the terminal character input ring buffer. This storage area is known as a character "silo" because it functions as a first-in-first-out holding buffer. The silo buffers are important in that they allow TSX-Plus to store characters that arrive in a burst. There are three parameters which set default values for the character silos: NCSILO, NCXOFF, and NCXON. The SILO macro may be used within a line definition block to specify silo parameters which override the default values for a specific line.

The NCSILO parameter specifies the default silo size. A value of 32 (decimal) is a reasonable silo size for most applications although you may want to use larger silos on lines that receive high-speed input from external devices or other computers. The character silo buffers are allocated in the 40Kb low-memory portion of TSX-Plus so you should not be wildly liberal in their allocation.

NCXOFF When a character silo fills to the point that only NCXOFF free character positions remain, the system will transmit an XOFF (ctrl-S) to try to stop transmission from the external device. The choice for NCXOFF depends on the speed of transmission, the time it takes for the XOFF to propagate to the sending device, and the time it takes for the sending device to respond to the XOFF and stop transmitting. A reasonable range of values for NCXOFF is 4 to 16. If characters

appear to be lost due to input silo overrun, increase the value of NCXOFF. The SILO macro may be used within a line definition block to specify the XOFF control parameter for a particular line.

NCXON — If the system sends an XOFF character because a silo buffer becomes nearly full, it will send an XON (ctrl-Q) to restart transmission when there are only NCXON characters remaining in the silo. The value of this parameter is not very critical; the recommended value is 4. The SILO macro may be used within a line definition block to specify the XON control parameter for a particular line.

CLXTRA — This parameter specifies the number of unattached Communication Line (CL) units to be in included in the system. These CL units are initially not assigned to any line but may "take over" an inactive time-sharing or CL line by use of the SET CL keyboard command or a TSX-Plus system service call (EMT). The total number of CL units included in the system equals the value of the CLXTRA parameter plus the value specified for the fourth parameter to the TBLDEF macro. No more than 8 total CL units may be included in the system. CL units are referenced as I/O devices using the names "CL0:", "CL1:", ..., "CLn:", where n is the total number of CL units minus 1.

CLORSZ — Each CL unit has an output character ring buffer which is used to transfer characters from the output data buffer to the communications controller. The output ring buffer allows the system to overlap the transmission of the final characters for one write operation with the time needed to get the next write operation started. The CLORSZ parameter specifies the default output ring buffer size for CL units. The recommended value for CLORSZ is ((3*baud_rate)/1000+2). For example, the recommended value for a 9600 baud line is ((3*9600)/1000+2) or 31. The BUFSIZ macro can be used within a CL line definition block to specify the output buffer size for an individual CL line.

NRMFLG — The NRMFLG parameter is used to specify a default set of options to apply to all lines unless the FLAGS macro is used within a line definition block. The value specified for the NRMFLG parameter must be the logical sum of a set of option symbols. When more than one option is specified, the names of the option symbols should be joined together with exclamation marks ("!") which is the MACRO assembler syntax symbol for the logical OR operation. The valid option symbols that may be used to form the value for NRMFLG are described below:

Flag	Meaning when set
$SCOPE	Terminal is a CRT type terminal and DELETE is to echo as backspace-space-backspace.
$ECHO	Echo characters to the terminal.
$TAPE	If this flag is set the line will be placed in "TAPE" mode. This mode of operation is useful if the line is receiving input from a paper-tape reader, cassette tape, floppy disk or another computer that is transmitting lines of data terminated by carriage-return and line-feed. When tape mode is selected, the system discards line-feed characters. TAPE mode can also be controlled by use of the SET TT [NO]TAPE keyboard command and the "W" and "X" program controlled terminal option functions.
$8BIT	If this flag is set, 8 bits of each received character are passed to the program. If this flag is not set, only the low order 7 bits of each character are passed to the program. Regardless of the setting of this flag, the null character (000) is never passed to the program. Also, if the VTxxx escape sequences are declared to be a TSX-Plus activation condition, the character 377 (octal) also may not be received. If 8 bit mode is selected, the terminals should be set to transmit and receive 8 bit characters without parity. The SET TT [NO]EIGHTBIT keyboard command can also be used to control this option.
$START	If this flag is set the line will be automatically initiated when TSX-Plus is started. If the flag is not set, the line will not be initiated until carriage-return is pressed at the terminal.
$NODET	If this flag is set the line is prevented from using the DETACH keyboard command which controls detached jobs.
$TAB	Do not simulate tabs by inserting spaces. Use with terminals whose hardware responds to tab characters, such as VT100 terminals.
$FORM	Do not simulate form feed by inserting line feed characters. Use with terminals whose hardware responds to form feed characters, such as LA120 terminals.
$AUTO	The $AUTO flag enables automatic baud rate selection (autobaud) for the line. This may only be used with lines connected to hardware controllers that support programmable speed selection such as DZ(V)11, DH(V)11, and DLV11-E. When autobaud is specified for a line the line's speed is

set to match the speed of the terminal connected to the line. When the carriage return character is received to start up a line, the system examines the (possibly garbled) character and uses the received character value to determine the speed of the transmitting terminal. The line's speed is then set to that value and the line is started. The following baud rates are supported by automatic speed selection: 110, 300, 1200, 4800, 9600, and 19200. It is necessary to type two consecutive carriage returns to start a line with autobaud if the speed is less than 1200 baud. A speed of 19200 is not supported by some hardware controllers such as the DEC DZ(V)11. The SET TT [NO]AUTOBAUD keyboard command can also be used to control this option.

$LC — Enables lower case input from the terminal. Note that bit 14 of the job status word must also be set to enable lower case input.

$NOVLN — If this flag is specified, the line will not be allowed to use the TSX-Plus virtual line facilities. That is, the line will always be connected to its primary line.

$DEFER — If this flag is set, "deferred" character echoing will be enabled. If the flag is not set, "immediate" character echoing will be used. See the description of the DEFER option to the SET TT command in the TSX-Plus Reference Manual for an explanation of deferred character echoing. It is recommended that deferred echoing mode be used.

$QTSET — If this flag is set the line will be initialized as if a "SET TT QUIET" command had been executed. This inhibits the listing of command files.

$PRIV — If this flag is set the line will be authorized for "operator privilege". See the section on operator privilege in the System Manager's Guide for an explanation.

$PHONE — This flag should be set if the line is connected to a dial-up telephone modem. If this flag is set, TSX-Plus will perform modem control such as answering the phone when the ring signal occurs and hanging up when carrier is lost. See the TSX-Plus System Manager's Guide for further information about modem control.

Note that the FLAGS macro can be used to specify options on a line-by-line basis and the SET TT keyboard command can be used to alter the settings of these options after the system is running.

3.3.9.2 Specifying the number of lines. Each line that is to be used as a TSX-Plus time-sharing line must be declared in TSGEN. The total number of lines is first declared by setting the proper values as arguments to the TBLDEF macro. The form of the TBLDEF macro is:

```
TBLDEF    num_real,num_virtual,num_detached,num_cl
```

The TBLDEF macro has four arguments:

1. The number of real (physical) time-sharing lines. This value should equal the number of LINDEFs which follow.

2. The number of virtual time-sharing lines.

3. The number of job slots to allocate for the execution of detached jobs.

4. The number of dedicated communication lines (CL lines). This value should equal the number of CLDEFs that follow.

See Chapter 4 of the TSX-Plus Reference Manual for more information on virtual lines and detached jobs.

TSX-Plus will support up to 31 total lines, including time-sharing lines, CL lines, virtual lines, and detached jobs. However, we do not recommend that you enable that many unless you actually plan to use all of them. The memory that is used by these lines is limited, and is shared with many other facilities. Therefore, you should not define more lines than necessary for your system. Performance with a large number of lines generated is highly dependent on configuration, specifically: CPU speed, I/O devices, and types of applications (programs) being run.

Refer to the example at the end of this section and to the examples in the supplied TSGEN module as you read the following explanation.

The actual line definitions follow the invocation of the TBLDEF macro. Each line definition is specified by creating a Line Definition Block (LDB). There must be exactly as many LDB's as there are physical lines and dedicated CL lines. Virtual lines and detached jobs are not described by LDB's.

A Line Definition Block for a time-sharing line begins by using the LINDEF macro and ends by using the LINEND macro. A line definition block for a dedicated CL line begins by using the CLDEF macro and ends by using the CLEND macro. Each LDB must have matching calls to LINDEF and LINEND or CLDEF and CLEND. Other optional macros may be used within the line definition block to specify parameters for the line.

TSX-Plus supports lines connected to DL11 and DLV11 serial communication cards and lines connected to DZ11, DZV11, DH11, and DHV11 multiplexers. TSX-Plus will support a mixture of dial-up and direct connect lines. Unless otherwise

stated, there is no distinction between DL11 and DLV11 support, nor between DZ11 and DZV11 support. However, DH11 support is different from DHV11 support.

3.3.9.3 Specifying multiplexer lines. The line definition blocks for lines connected to DZ11, DH11, and DHV11 multiplexers are enclosed within a Multiplexer Definition Block (MDB). An MDB begins with a DZDEF, DHDEF, or DHVDEF macro call, contains the LDB's for all lines connected to the multiplexer, and ends with a MUXEND macro call. Thus the general model for defining a group of lines connected to a multiplexer is:

```
DZDEF or DHDEF or DHVDEF    ;Start of lines for multiplexer
LINDEF                      ;Start first line definition block
...                         ;Parameters relating to first line
LINEND                      ;End of first line definition block
LINDEF                      ;Start of next line definition block
...
LINEND                      ;End of line definition block
MUXEND                      ;End of lines for this multiplexer
```

The DZDEF macro (for DZ11 multiplexers) and the DHVDEF macro (for DHV11 multiplexers) both require two arguments and are similar in form:

```
DHVDEF    vector,csr

DZDEF     vector,csr
```

The first parameter ("vector") specifies the address of the receiver interrupt vector, the second parameter ("csr") specifies the address of the Control and Status Register (CSR).

The DHDEF macro is used to specify values for a DH11 multiplexer. It requires four parameters and has the following form:

```
DHDEF     dhvector,dhcsr,dmvector,dmcsr
```

The first parameter ("dhvector") is the address of the receiver interrupt vector for the DH11, the second parameter ("dhcsr") specifies the address of the Control and Status Register (CSR) for the DH11. The third parameter ("dmvector") specifies the address of the interrupt vector for the associated DM11 modem control unit, and the fourth parameter ("dmcsr") specifies the address of the CSR register for the DM11. A DM11 is an optional unit which provides modem control for DH11 lines. If there is not a DM11 associated with the DH11, specify 0 (zero) for the third and fourth parameters.

Note that the system interface to a DH11 multiplexer is different from that for a DHV11 multiplexer (they have a different control register structure). Thus a DHV11 is not the same as a DH11 compatible multiplexer installed on a Q-bus system. Therefore, choose the DHDEF or DHVDEF macros based on the type of multiplexer, not on the type of system on which they are installed.

TSX-Plus uses the DMA output capabilities of the DH11 and DHV11 multiplexers which reduces system overhead for character output by about 14% compared with a DZ11 driving a time-sharing line or about 20% for a CL line.

TSX-Plus will support up to four multiplexers (but not more than 31 total lines, including virtual lines and detached jobs). If there is more than one multiplexer, each must be defined using a separate multiplexer definition block beginning with a DZDEF, DHDEF, or DHVDEF macro and ending with a MUXEND macro. Note that the multiplexer port numbers (as specified with the LINDEF or CLDEF macros for multiplexer lines) start over at 0 for each multiplexer.

3.3.9.4 DL11 line definitions. Line definition blocks for lines connected to DL11 type controllers are somewhat simpler than multiplexer line definition blocks. In the case of DL11 lines, the addresses of the DL11 interrupt vector and Control and Status Register (CSR) are specified with the LINDEF macro that starts the line definition block.

Different model DL11 cards have different ranges of addresses for the status register. DL11-A and B cards generally start at 176500, while DL11-C, D, and E cards start at 175610. The addresses increase by 10 (octal) per line. Note that 16-bit device addresses are specified in TSGEN. The receiver interrupt locations for DL11 cards normally start at 300 and increase by 10 (octal) per line.

Note on DLV11-J controllers, port 3 is special and can be configured by wirewrap jumper to have its vector and CSR address at the normal operators console location (60 and 177560) or at a location above the addresses for port 2. If you have more than one DLV11-J controller make sure you do not have more than one line configured as the console terminal.

The receiver status register and interrupt vector addresses for all devices are normally written on a card that is attached to the top cover of the CPU drawer for UNIBUS machines and somewhere in the cabinet for Q-bus machines. If you cannot locate the status register and interrupt vector addresses, default addresses may be found in the PDP-11 Programmer's Reference Card or the processor handbook for your machine. If you still are having problems, contact the person who installed your machine.

The most common problem in getting started with TSX-Plus is specifying incorrect addresses for the communication cards. Make sure the vector and CSR addresses of your time-sharing line controllers do not conflict with those for other devices such as LS or LP.

3.3.9.5 Time-sharing line definition blocks. The line definition block for a time-sharing line begins with a LINDEF macro and ends with a LINEND macro. The LINDEF macro requires different parameters depending on whether it describes a DL11 type line or a multiplexer line. The LINDEF macro for a DL11 type line requires two parameters and has the form:

```
LINDEF    vector,csr
```

The first parameter ("vector") is the address of the input (receiver) interrupt vector. The second parameter ("csr") is the address of the receiver control and status register (CSR).

A LINDEF macro for a multiplexer line requires only a single argument and has the form:

```
LINDEF    muxport
```

Where "muxport" is the number of the port within the multiplexer group. DZ11 lines are numbered 0 to 7 and DZV11 lines are numbered from 0 to 3. DH11 lines are numbered 0 to 15, and DHV11 lines are numbered 0 to 7. Note that a decimal point should be specified with the line number to indicate that the value is decimal.

The LINDEF macro also accepts a third (second for multiplexer lines) optional parameter. One terminal may be declared to be the operator's console that receives system control messages such as requests for special form mounts if spooling is used. The terminal to be the operator's console is signified by specifying "OPER" as the third argument. Only one terminal may be declared to be the operator's console.

3.3.9.6 CL line definition blocks. The line definition block for a CL line begins with a CLDEF macro and ends with a CLEND macro. The CLDEF macro requires different parameters depending on whether it describes a DL11 type line or a multiplexer line. The CLDEF macro for a DL11 type line requires three parameters and has the form:

```
CLDEF     unit,vector,csr
```

The first parameter ("unit") specifies the CL I/O device unit number by which this line will be referenced. For example, unit number 3 is referred to by the name "CL3:". The units may be specified in any order but the unit numbers must be less than the total number of CL units ((fourth TBLDEF parameter) + CLXTRA). The CL unit numbers must also be unique. The second parameter ("vector") is the address of the input (receiver) interrupt vector. The third parameter ("csr") is the address of the receiver control and status register (CSR).

A CLDEF macro for a multiplexer line requires two parameters and has the form:

```
CLDEF     unit,muxport
```

Where "unit" is the CL unit number and "muxport" is the number of the port within the multiplexer group.

The BUFSIZ, SPEED, and FLAGS macros may be used within a CL line definition block to specify parameter values for the line.

3.3.9.7 Line definition control macros. Optional macros may be invoked between the LINDEF and LINEND calls to set parameters for a line. The available macros are listed below.

FLAGS This macro is used to set a variety of control flags for the line. The form of the FLAGS macro is:

```
FLAGS     flag_values
```

The single argument to FLAGS must be the logical sum of those flags that are to be set for the line. The symbolic flag names are the same as specified for the NRMFLG parameter (see above). When more than one flag is specified, the names of the flags should be joined together with exclamation marks ("!") which is the MACRO assembler syntax symbol for the logical OR operation.

The FLAGS macro may be used within the line definition block for a CL line but only the following flags are significant: $8BIT, $TAB, $FORM.

Note that the FLAGS macro sets the default line characteristics when each line is started and that the "SET" keyboard command may be used to alter flag settings for a line. Thus, the command "SET TT LC" would enable lower case input, and "SET TT NOSCOPE" would say that the terminal is not a CRT device.

SPEED The SPEED macro is used to specify the transmit/receive baud rate for lines connected to hardware controllers that support programmable baud rates such as DLV11E, DZ11, DH11 and DHV11. The form of the SPEED macro is:

```
SPEED     speedcode
```

The SPEED macro requires a single argument which specifies the line's operating baud rate. The following codes may be used: S75, S110, S135.5, S150, S300, S600, S1200, S1800, S2000, S2400, S3600, S4800, S7200, S9600, and S19200. That is, the speed parameter should be specified as a symbolic name formed from concatenating the letter "S" with the baud rate.

A baud rate of 19200 is not supported by DL11 and DZ11 lines. Baud rates of 2000, 3600, and 7200 are not supported by DH11 lines. Baud rates of 3600 and 7200 are not supported by DHV11 lines.

If a line definition block does not contain a SPEED macro, the speed specified for the last defined line will be used.

The SPEED macro can be used in line definition blocks for time-sharing lines and CL lines.

TRMTYP The TRMTYP macro is used to declare what type of terminal will be used with the line. The form of the TRMTYP macro is:

```
TRMTYP    terminal
```

The terminal types DIABLO and QUME are treated as equivalent and no longer support the ETX/ACK protocol which was available with earlier versions of TSX-Plus. Note that newer terminals which use the X-ON/X-OFF (DC1/DC3, CTRL-Q/CTRL-S) protocol are acceptable. See the SET TT command in Chapter 2 of the TSX-Plus Reference Manual for more information on the meaning of each terminal type. The valid choices are listed below:

Name	Terminal type
VT200	DEC VT200 family terminal
VT100	DEC VT100 family terminal
VT52	DEC VT52 terminal
LA36	DEC LA36 terminal
LA120	DEC LA120 terminal
HAZEL	Hazeltine brand terminal
ADM3A	Lear Siegler ADM3A terminal
DIABLO	Diablo brand terminal (with X-ON/X-OFF protocol)
QUME	Qume brand terminal (with X-ON/X-OFF protocol) (Diablo and Qume are treated as equivalent)

BUFSIZ The BUFSIZ macro is used to specify the number of characters to reserve for the line's input character ring buffer (argument 1) and the output character ring buffer (argument 2). The form of the macro is:

```
BUFSIZ    inputsize,outputsize
```

If a BUFSIZ macro is not used in a Line Definition Block, the default sizes as specified for DINSPC and DOTSPC will be used.

If the BUFSIZ macro is used within a CL line definition block, the first argument (input buffer size) is ignored but the second argument (output buffer size) is used to control the size of the CL unit output buffer. If no BUFSIZ macro is used within a CL line definition block, the value of the CLORSZ parameter controls the size of the output ring buffer.

SILO The SILO macro is used to specify values for parameters related to character input silo buffers. The NCSILO, NCXOFF, and NCXON parameters (see above) specify default silo values which will apply to all lines unless a SILO macro is used within a line definition block to specify values for a specific line. The form of the SILO macro is:

```
SILO ncsilo,ncxoff,ncxon
```

The three parameters specify the silo size, the number of free character positions remaining at which point an XOFF is transmitted, and the number of remaining characters at which point an XON is transmitted to restart the sender.

A SILO macro may be used in line definition blocks for both time-sharing and dedicated CL lines. Space may be saved by specifying a silo size of zero for CL lines that will be used for output only (e.g., a CL line used to drive a printer).

CMDFIL The CMDFIL macro is used to specify the name of a start-up command file to be executed when the line is initialized. The form of the macro is:

```
CMDFIL    dev:file.ext
```

This macro has one argument that is the name of the command file (dev:file.ext). This argument must be included in order to use the TSX-Plus LOGON facility.

The CMDFIL macro should not be used in the line definition block for a dedicated CL line.

This ends the description of macros that can be used within Line Definition Blocks.

3.3.10 Line definition example

The following example shows the definition of two DL11 lines and three DZ11 lines. Two virtual lines are also declared (LDB's are not used for virtual lines). Note that the exclamation mark is used to perform the logical OR (sum) operation when combining flags.

```
NRMFLG  =  $ECHO!$DEFER!$LC

        TBLDEF  5.,2.,1.,1.             ;5 real, 2 virtual, 1 detached, 1 CL

; Define DL11 lines
; Define DL11 line #1
        LINDEF  300,175610      ;DL11-E line. Vector=300, CSR=175610
        TRMTYP  VT200           ;Terminal type
        SPEED   S9600           ;Speed = 9600 baud
        FLAGS   NRMFLG!$SCOPE   ;Control flag options
        BUFSIZ  120.,300.       ;Input and output ring buffers
        CMDFIL  SY:START.COM    ;Start up command file
        LINEND                  ;End of DL11 line definition block
; Define DL11 line #2
        LINDEF  60,177560,OPER  ;Console terminal
        TRMTYP  VT200           ;Terminal is a VT200
        SPEED   S9600           ;9600 Baud
        FLAGS   NRMFLG!$PRIV!$START ;Privileged line with auto startup
        LINEND

; Define DZ11 lines
        DZDEF   310,177620      ;Start of DZ11 definition block
; Define DZ11 line # 0
        LINDEF  0               ;MUX line #0
        TRMTYP  VT100           ;Terminal type
        SPEED   S1200           ;1200 baud
        FLAGS   NRMFLG!$SCOPE
        CMDFIL  SY:LINE1.TSX
        LINEND
; Define DZ11 line # 1
        LINDEF  1               ;MUX line #1
        SPEED   S300            ;300 baud
        TRMTYP  LA36
        CMDFIL  SY:LOGON.TSX
        LINEND
; Define DZ11 line # 6
        LINDEF  6               ;MUX line #6
        TRMTYP  VT100
        SPEED   S9600           ;9600 baud
        FLAGS   NRMFLG!$PHONE!$AUTO ;Phone line with autobaud speed select
        CMDFIL  SY:LINE5.TSX
        LINEND
```

```
; Define DZ11 line # 7 as CL line 0
        CLDEF   0,7             ;MUX line #7 is CL0:
        SPEED   S4800           ;4800 baud
        FLAGS   $FORM           ;Printer supports form feeds
        CLEND
; End of MUX lines
        MUXEND                  ;End of MUX definition block
```

3.3.11 Defining start-up files for detached Jobs

The DETACH macro may be used to specify the names of start-up command files to be initiated as detached jobs when the TSX-Plus system is started. The use of the DETACH macro should follow the last line definition block. The form of the DETACH macro is:

```
        DETACH  dev:file.ext
```

The DETACH macro requires one argument which is the name of the command file to be initiated as a detached job. The physical device name must be included in the command file specification.

There may be one use of the DETACH macro for each detached job slot specified with the TBLDEF macro. If there are more detached job slots defined than there are invocations of the DETACH macro, the excess job slots are left idle when the system is started and detached jobs may be started on these lines by use of the DETACH keyboard command.

4. LINKING AND STARTING TSX-Plus

4.1 Assembling the modified TSGEN module

Once the TSGEN module has been modified to contain the desired parameter settings, it must be assembled using MACRO. The command to do this is:

```
.MACRO TSGEN
```

If you want to get a listing of the modified TSGEN (a good idea), use the command:

```
.MACRO/LIST TSGEN
```

No errors should occur during the assembly.

4.2 Linking TSX-Plus

The final stage of building TSX-Plus is to link the component parts together. A command file to do this is provided on the TSX-Plus distribution disk with the name "TSXLNK.COM". The system command to execute this command file is

```
.@TSXLNK
```

This command file creates three SAV files: TSX.SAV, TSKMON.SAV, and SYSMON.SAV. The TSX.SAV and TSKMON.SAV files must be on the system disk (SY:) before TSX-Plus can be started. The SYSMON.SAV file is only needed if the SYSMON dynamic system display utility program is to be used (see the System Manager's Guide for information about SYSMON). Note that the RTVECT sysgen parameter must have a value of 1 or greater if SYSMON is to be used. Both TSX-Plus and TSKMON must be rebuilt if any parameters are changed in TSGEN.

The linker will print the following warning message while linking TSKMON:

```
?LINK-W-Multiple definition of $OVRH
```

This is expected and is not an error.

Warning: Do not relink TSKMON onto the system device or copy it there while running under TSX-Plus. The position of the TSKMON file on the system disk must not change while TSX-Plus is running.

4.3 Starting TSX-Plus

Before starting TSX-Plus, you should check the following items to make sure the system is set up correctly:

1. The TSX.SAV, TSKMON.SAV, and CCL.SAV files must be on the system disk (SY:). CCL.SAV is provided on the TSX-Plus distribution disk; TSX.SAV and TSKMON.SAV are built using the TSXLNK.COM command file as part of the system generation process.

2. The TSX-Plus version of device handlers (with the extension ".TSX") must be on the system disk for each device declared in TSGEN.

3. The RT-11 device handlers for the devices on which the TSX-Plus swap and spool files will be placed must be installed (but need not be loaded) in the running version of RT-11.

4. If the TSX-Plus logon facility is being used, the LOGON.SAV and ACCESS.TSX files must be on the system disk. (The ACCESS.TSX file is created by the TSAUTH account authorization program.) If the account authorization file was created with a version of TSAUTH prior to the one supplied with version 4.0, use the AUTCVT program to convert the format of the account authorization file.

5. If user-defined commands are allowed, the TSXUCL program must be on the system disk.

6. Any startup command files associated with time-sharing lines (such as LOGON.TSX or LINE1.TSX) must be on the system disk.

7. If any shared run-time systems were specified in the system generation, they must be on the appropriate disks before the system is started.

8. TSX-Plus must be started under RT11SJ (single job); an attempt to start it under the foreground-background (FB) or extended-memory (XM) versions of RT-11 will result in a "?KMON-F-Insufficient memory" error message. This message may also occur if RT-11SJ has been sysgenned to include multi-terminal support or other features which make it too large to be able to start TSX-Plus, or if the USR is set NOSWAP or if too many device handlers are loaded. If your RT-11 has been sysgenned to include multi-terminal support, copy and boot the distributed versions of RT11SJ or RT11BL, then run TSX-Plus.

Once you have determined that all of these conditions are met, you can start TSX-Plus by typing

```
R TSX
```

After this is typed, time-sharing lines that were generated to be automatically starting ($START flag) should start up and print the TSX-Plus greeting message. Other lines will be initiated when carriage return is pressed at the terminal.

During its initialization TSX-Plus performs a test to make sure the physical lines defined in TSGEN actually exist. It does this by trying to access the receiver status register for each line. If a trap occurs, and the INIABT flag is set to 1, TSX-Plus displays the message:

```
?TSX-F-Invalid status register address for T/S line: xxxxxx
Line # = nn
```

If this occurs you must edit in the correct line address in the TSGEN module. TSX-Plus does not check the interrupt vector addresses for the lines, so if the system dies when a line is started, check to see if its interrupt vector address is correctly specified.

When TSX-Plus is running, the system line frequency clock must be operating at all times. This is true even if only a single job is being run under TSX-Plus.

If TSX-Plus does not start properly, carefully review the parameter settings in TSGEN. Check especially the values provided for DL11 interrupt vectors and receiver status registers. Different models of DL11 cards use different addresses.

In the case where RT-11 runs successfully on a system but TSX-Plus does not, look carefully at the memory installed on the machine above 56Kb. If it is not functional or improperly configured, TSX-Plus will not run. If you are using non-DEC peripherals, check with the peripheral vendor to make sure the device can support extended memory addressing. Refer to the TSX-Plus System Manager's Guide for information on error messages received during TSX-Plus startup and operation.

4.4 Setting the memory allocation for system programs

SETSIZ.SAV is a program that can be used to store information in a SAV file about how much memory TSX-Plus should allocate for the program when it is run. The method used to store this information in the SAV file does not affect the execution of the program when being run under RT-11. See Appendix A of the TSX-Plus Reference Manual for complete information about the SETSIZ program.

A command file named SETSIZ.COM is provided with TSX-Plus. It contains the necessary commands to cause the SETSIZ program to set appropriate allocation sizes for most of the commonly used system programs. To execute this command file, make sure the SETSIZ.SAV program and SETSIZ.COM command file is on the system disk then type:

```
@SETSIZ
```

The allocation sizes set by this command file should be adequate for most sites but a particular site might wish to alter them based on special requirements. Note that it is not necessary to execute the SETSIZ.COM file every time TSX-Plus is started since the size information is stored permanently in the program SAV files.

5. PATCHING AND BUILDING TSX-Plus DEVICE HANDLERS

5.1 Device Handlers for TSX-Plus

TSX-Plus generally uses standard RT-11 XM device handlers, however, the DD, DL, DM, DX, DY, MM, MS, and MT handlers as supplied with RT-11 require minor modifications to function correctly with TSX-Plus. The necessary handler modifications are supplied as SLP files with TSX-Plus. These SLP files have already been applied and are included in the dd.TSX handlers supplied with TSX-Plus.

If you ordinarily need to make no modifications to the handlers supplied by Digital on your system, then you may use the handlers provided with the TSX-Plus distribution. Most common changes can be accommodated through device SET options. However, if you need to change the handlers supplied with RT-11, you may need to apply some patches before using them. An example of a change that requires regenerating a device handler is adding a second controller (vector and CSR) to the MT handler.

5.2 Patching device handlers

Handlers which do not require patches for use with TSX-Plus are: CT, CR, DL, DP, DS, DT, DU, LP, LS, NL, PC, RF, and RK. Some device handlers (DD, DL, DM, DX, DY, MM, MS, and MT) require minor modifications to execute properly with TSX-Plus. When using the file structured magtape device handlers, the file structured module (FSM) as well as the device specific handlers (TJ, TM, and TS) must be patched. The dd.TSX device handlers provided with TSX-Plus have already been patched using the SLP files provided on the distribution media. These patched handlers have been assembled with the appropriate conditionals (ERL$G=0; MMG$T=1; TIM$IT=1) and linked to create the dd.TSX files provided with TSX-Plus. Device handlers do not need to be rebuilt unless you require some modification which can not be accomplished with a device SET option.

If it is necessary to rebuild a device handler, copy the SLP files from the TSX-Plus distribution media. The SLP files have the extension .SLP. Copy the RT-11 handler source, named dd.MAC, from the RT-11 distribution media to dd.OLD on a scratch working pack (e.g., COPY DL1:DM.MAC DL0:DM.OLD). Apply the patch using a command of the form:

```
SLP dd.MAC=dd.OLD,ddxxx.SLP
```

where "dd" is the two character source file name.

For more information on the use of SLP, refer to Chapter 21 of the RT-11 System Utilities Manual. Note: Do not apply the patch to the original RT-11 distribution media as any RT-11 patches will require the original source file.

5.3 Building device handlers

When building device handlers, it is necessary to set certain switches before assembly which control conditional code exclusion and inclusion. TSX-Plus requires memory management and optionally allows device timeout. However, it does not support error logging, therefore, error logging should not be specified when the handlers are built.

An easy method of building device handlers for TSX-Plus is to create a TSX-Plus conditional file. Using the editor, create a file "TSXCND.MAC" with the following conditionals:

```
MMG$T   = 1     ;enable memory management
ERL$G   = 0     ;disable error logging
TIM$IT  = 1     ;optionally enable timeout
```

Note that setting a conditional parameter to zero (0) disables the option and setting it to one (1) enables the option. Since device timeout is optionally supported, TIM$IT may be either 0 or 1. Other parameters may be included to specify device characteristics. For instance, the following conditionals may be specified for RL01/RL02 support:

```
DL$UN   = 2      ;define 2 units for the DL handler
DL$CSR  = 176400 ;define the CSR address for the DL handler
DL$VEC  = 164    ;define the vector for the DL handler
```

The following conditionals might be specified for file structured MT support:

```
MT$FSM  = 1      ;enable file structured MT support
MT$UN   = 2      ;define 2 units for the MT handler
MT$CSR  = 172520 ;define the CSR address for the MT handler
MT$VEC  = 224    ;define the vector for the MT handler
```

Refer to Appendix C of the RT-11 System Generation Guide for an entire list, default value, and description of device conditionals. These parameters are not required and will use a default value if left unspecified, except MMG$T which must be set to 1 for TSX-Plus.

Whether or not patching is required, most handlers may be built (when necessary) by the following commands:

```
MACRO  TSXCND+dd/OBJ
LINK/EXE:SY:dd.TSX  dd
```

where "dd" represents the two character device name.

Only the file structured magtape handlers require different commands. They may be built by using the following commands:

```
MACRO  TSXCND+FSM/OBJ
MACRO  TSXCND+td/OBJ
```

```
LINK/EXE:SY:dd.TSX  td,FSM
```

where "td" represents the tape device source module name (TJ, TS, or TM) and "dd" represents the corresponding magtape device name (MM, MS, or MT). Notice that the LINK command automatically appends the "TSX" file extension. Since TSX-Plus uses handlers with the extension "TSX", the handlers must be linked with that extension rather than with the extension "SYS". This allows the TSX-Plus handlers to coexist on the same system disk with standard RT-11 handlers without conflict. Handlers for all devices included in your TSGEN DEVDEF list, including the system disk, must be on the system disk when TSX-Plus is started.

6. TSX-Plus Installation and System Generation Hints

Whether using the short-cut installation process or the complete system generation process, various types of errors can occur and be reported during the installation process. The errors discussed in this section are those most commonly encountered when first installing TSX-Plus.

A complete listing of start-up error messages which may be reported by TSX-Plus are included in an appendix of the TSX-Plus System Manager's Guide. The System Manager's Guide also lists those fatal system error messages which may be reported at any time during the operation of TSX-Plus; these are usually indicative of an unusual hardware condition.

If an error is reported during start-up of TSX-Plus which is not described in those two appendices, it will have originated from RT-11 before TSX-Plus took control; see the RT-11 System Message Manual for descriptions of RT-11 error messages.

Error message: ?PIP-F-Device full

There is not enough free space on the output disk to copy all distributed files. The system disk requires about 2000 free blocks: about 400 blocks for the executable files and handlers, about 1100 blocks for the swap file, and an additional 500 blocks for the spool file. The working disk requires about 1400 free blocks: about 800 blocks for distributed source and object files, and about 800 blocks for files created while generating TSX-Plus. Remove unnecessary files from the system and working disks and squeeze if necessary.

Error message: ?KMON-F-Not enough memory (RT-11 V4)
?KMON-F-Insufficient memory (RT-11 V5)

The base of the monitor is too low to load the TSX.SAV program. Unload some resident device handlers and SET USR SWAP. TSX-Plus may not be started from the RT-11FB or RT-11XM monitors. The TSX.SAV program must be run from the RT-11 Single-Job or Baseline monitor which has NOT been "sysgenned" to include multi-terminal support. Copy the original RT-11 Single-Job monitor (RT11SJ.SYS) from your RT-11 distribution and boot it before running TSX-Plus.

Error message: ?TSX-F-(*** error message displayed here ***)

See the TSX-Plus System Manager's Guide for descriptions and remedies for fatal errors which occur during start-up of TSX-Plus.

No response from terminals:

The operator console is established with vector=60 and CSR=177560. This and all other time-sharing terminals must have their hardware interface vector and CSR address settings declared to TSX-Plus by parameters in the file TSGEN.MAC before assembling and linking the TSX-Plus system. Only DL(V)11 type interfaces are usable with the short-cut installation method; DZ(V)11 and DH(V)11 multiplexers require that you perform a TSX-Plus system generation. The short-cut installation process assumes that time-sharing lines are configured as follows:

TSX-Plus Line #	Vector	CSR
1	60	177560
2	310	176510
3	320	176520

Error message: ?TSX-F-Fatal system error @nnnnnn
(EEE-error message displayed here)
Arg. value = xxxxxx
Seg. value = yyyyyy

The most common causes of this error are incorrect device vector and address specifications. If the vector is incorrect, the error message will be: "UEI-Interrupt occurred at unexpected location". If the address is incorrect, the error will be: "KTP-Kernel mode trap". Review the default time-sharing line vectors and addresses above. The only other devices declared to TSX-Plus in the short-cut installation method are the following:

Device	Vector	CSR
DL	160	174400
DM	210	177440
DU	154	172150
DY	264	177170
LP	200	177514
NL	N/A	N/A
RK	220	177400

The vector and address assigned to the printer may be changed using the keyboard "SET" command after TSX-Plus has been started, but before printing anything; see the next paragraph. See the TSX-Plus System Manager's Guide for explanations of other fatal error messages.

Printer does not respond:

The interface hardware for the printer must correspond to the address specified in the device handler LP.TSX. If your printer is not at the default vector and address, then after starting TSX-Plus, but before printing anything, use the keyboard handler "SET" options to set the correct vector and address.

```
SET LP VECTOR=nnn
SET LP CSR=nnnnnn
```

The printer vector and address must not conflict with those defined for any time-sharing terminal line. After setting the correct vector and address in the LP.TSX file, reboot the system and restart TSX-Plus with the command:

```
.R TSX
```

Appendix A - System Size Calculation

A.1 System size and sysgen features

The TSX-Plus system is divided into two portions: an unmapped portion that consists of kernel code, device handlers, and job control tables; and a mapped portion that consists of virtual overlays for the monitor, shared run-time systems, and data areas such as data caching buffers and time-sharing terminal character buffers. The unmapped portion of the system is constrained to 40Kb. The mapped portion is only constrained by the physical memory installed on the system and the amount of memory that needs to be made available for time-sharing jobs.

The following table indicates the number of bytes of code and/or data space added to the mapped and unmapped portions of the system by various system features.

Effect of System Components on Overall System Size

System Component	Bytes in Unmapped Region	Bytes in Mapped Region
Each additional time-sharing line	850 + input character silo buffer	Terminal input character buffer + output character buffer
CL Handler	150	2224
Each dedicated CL line	382 + output ring buffer	0
Each extra CL unit	30 + output ring buffer	0
DH-11 or DHV-11	980 + (num lines)*64	0
Unmapped handler	Size of device handler	0
Mapped handler	26	Size of device handler
Device spooling	18 + (spool file size)/8 + (number spool files)*24 + ((number spool dev.)* (45+2*(num. backup blk.)))	2100 +(num. spool buffers)*512

(continued)

System Component	Bytes in Unmapped Region	Bytes in Mapped Region
Shared file record locking and data caching	10 + NUMDC*8	2100 + NUMDC*512 +MAXSF*14 +(num. time share jobs)*6 +MAXSFC*(12+2*MXLBLK)
Generalized data caching	1940	CACHE*528
Directory caching	MAXCSH*14	NMFCSH*18
Inter-job message communication	0	1240 + (MAXMC+MAXMRB)*12 + MAXMSG*(4+MSCHRS)
PLAS support	0	2300
Real-time support	490 + RTVECT*10	1010
Single line editor	0	3300
Program debugger	0	6272
I/O mapping (18-bit device support on 22-bit Q-bus systems)	22 + MIONBF*16	960 + MIONBF*(MIOBSZ*512)
Performance analysis monitor	0	PMSIZE
Shared run-time	(num. of run times)*14	Size of run time system

A.2 Device Handler Sizes

The following table lists the size of the device handlers which are distributed with TSX-Plus.

Handler	Size (bytes)
CL	(See preceding table)
CR	778
CT	2322
DD	1252
DL	1304
DM	1316
DP	346
DS	246
DT	256
DU	826
DX	594
DY	738
LP	318
LS	630
MM	4308 (file struct.)
MS	4768 (file struct.)
MT	3914 (file struct.)
NL	58
PC	210
RF	232
RK	280
VM	306

Appendix B - Device CSR and Vector Address Table

System _____ CPU & Serial # _____

Memory installed _____ Peripherals installed _____

_____

Interface device(s) installed _____

_____

Device	Type	Handler	Vector	CSR/RSR	Speed	Parity	Stop-bits